SECURE FILE TRANSFER SYSTEM

This application claims the benefit of U.S. Provisional Application No. 60/203,746, filed 12 May 2000, which provisional application is incorporated by reference herein.

Technical Field

The invention relates to secure file transfers over computer networks, especially secure file transfers involving encryption of the file.

Background of the Invention

There are many encryption schemes available to computer users for secure file transfer, but most require that the user download a software application for encryption of the file before sending the file. Tumbleweed, in U.S. Patent number 5,790,790 to Smith et al., developed a less burdensome document delivery system that is used by many delivery companies to facilitate delivery of "e-packages," but the scheme suffers from drawbacks. One of the most significant drawbacks is the system's use of relatively weak encryption based on the Secure Sockets Layer, which cannot be changed without a fundamental alteration of the transfer scheme.

Summary of the Invention

The instant invention overcomes the drawbacks of the prior art by providing strong encryption in a relatively client-independent format using a client-side application, such as a Java applet run on the client side to encrypt the file, preferably using elliptical encryption. Further, the preferred embodiment uses a second encryption method to encrypt each block of the encrypted file as it is sent to the server by the client-side application, such as the applet previously mentioned, the server storing the blocks as they arrive and reassembling the encrypted file. The system notifies the

recipient of the presence of the file, preferably in an e-mail message or the like including a hypertext link; and the process is reversed when the recipient accesses the file.

Description of the Drawings

FIG. 1 is a schematic representation of the server, network, and clients used in the instant invention.

FIG. 2 is a schematic representation of the invention deployed in a server.

FIG. 3 is a schematic flow diagram of paths users can follow within the preferred embodiment of the invention as well as some actions taken by the system in response thereto.

FIG. 4 is a schematic flow diagram of a preferred implementation of the encryption features of the invention.

FIG. 5 is a schematic flow diagram of a key pair encryption scheme usable in the invention.

FIG. 6 is a schematic screenshot of a main secure file transfer page of a preferred implementation of the invention.

FIG. 7 is a schematic screenshot of a destination entry page of a preferred implementation of the invention with an addressee entered into the destination entry field.

FIG. 8 is a schematic screenshot of the destination entry page of a preferred implementation of the invention as shown in FIG. 7 with a user-addressee listed in the destination list after pressing the "Add" button in FIG. 7.

FIG. 9 is a schematic screenshot of the destination entry page of a preferred implementation of the invention as shown in FIG. 7 with an e-mail-addressee listed in the destination list after pressing the "Add" button in FIG. 7.

FIG. 10 is a schematic screenshot of a preferred implementation of the invention as the client machine receives the encrypter from the server.

FIG. 11 is a schematic screenshot of the encrypter of a preferred implementation of the invention prompting the user to identify a file for transfer on a volume to which the client machine has access.

FIG. 12 is a schematic screenshot of the encrypter of a preferred implementation of the invention notifying the user of an interrupted transfer.

FIG. 13 is a schematic screenshot of a page allowing designation of addressees and files for sending from a server-controlled storage medium under a preferred implementation of the invention.

FIG. 14 is a schematic screenshot of an inbox of the invention.

Description of the Invention

The instant system provides subscriber users with the ability to transfer strongly encrypted documents to other subscribers and to non-subscribers. The system tolerates transfer interruptions and, since it is based on Java technology, requires no software other than a conventional Java enabled Web browser. The steps the system undergoes can be broken down into a few well-defined actions. The system applies strong encryption to all files to provide the highest level of security for users, and the system maintains a history of all transfers to assist users in tracking senders and recipients.

The system can use recipient information from the Information Distribution System of U.S. Patent Application number _____ filed concurrently herewith and can be used with the Information Autocompletion System of U.S. Patent Application number _____ filed concurrently herewith. The disclosures of the above-mentioned two applications (_____ and _____) are hereby incorporated by reference.

Sending a Document

To send a document, a user visits the request page and provides a destination in the form of a subscriber username or non-subscriber e-mail address. The system allows the user to designate a path to the file the user wishes to transfer or to use conventional GUI dialog box technology to browse accessible storage media to locate and select the file to be sent. The system preferably includes a status display, initially set to "Ready" by default, so the user can easily tell how the transfer proceeds. When the user has provided the destination and file to be transferred, the user initiates transfer by, for example, clicking a "Send" button on the request page. I prefer to also provide an additional "Quick Send" option at this point. Once the user initiates transfer, the system begins breaking up and encrypting the file; and the system preferably provides a "Stop" button or the like to allow cancellation of the transfer.

The request page preferably displays a number of statistics for the user. For example, if users are given a limit on the number of free transfers they can make, the system can display how many transfers are left; if the system imposes a file size limit on the user, the system can display this as well. The system can also display user messages, such as how long the file will be stored on the system before deletion.

As the system uploads the file, an application on the client-side, such as a Java applet, breaks the file into blocks of a predetermined size. I prefer to use a fixed block size (10 KB, for example), but the block size can also be based on the size of the original file. The system then generates a request, which the system sends to the client-side application from the server-side application hosting the main portion of the system. The server-side application sends all parameters required for the encryption portion of the transfer; where the system uses elliptical encryption, the parameters will include all parameters (q, a, b, r, G) that define an elliptical curve (EC). The client-side application generates a shared, secret key (K) using, for example, the Mendez-Qu-Vanstone public key agreement scheme with cofactor multiplication according to IEEE P1363 draft

February 8, 1999, the disclosure of which is hereby incorporated by reference. The client-side application then encrypts the encrypted file block (FB) using a symmetric encryption algorithm with K, K(FB). The encrypted block, along with the key, is sent to the server and stored in the system database. The file can be "unsent" up to the time the recipient downloads the file.

At the receiving end, the recipient can download the file via a simple and intuitive process. The user simply opens a client-side application, such as a Java applet, that presents the user with a form including a download progress indicator, a destination field, an initiation object, and an abort object. The download progress indicator allows the user to easily monitor the status of the download at any particular time; as with the upload, the initial display is something along the lines of "Ready" by default. The destination field can be completed manually (typing in a destination path for the file) or by invoking a conventional GUI dialog box to browse accessible storage media to locate and select the destination. The system then sends the encrypted file in blocks of varying size, each block including its own key that accompanies the document. If a transfer error occurs, this method of transfer allows the user to resume download from the point of the error instead of starting over from the beginning of the document.

The preferred encryption algorithm for the encryption key of the instant invention is elliptical curve (EC) encryption. The client-side application, such as a Java applet, the user downloads from the server preferably includes all parameters required to define the elliptic curve used in the encryption; and the applet preferably generates a shared, secret key using the Mendez-Qu-Vanstone public key agreement scheme with cofactor multiplication. The key K is sent from the system database on a server; K is preferably encrypted with the elliptical curve, and the applet decrypts the encrypted key KEC(K) using KECK=KEC(KEC(K)). Once the applet decrypts the key K, the applet sends a confirmation to the server and requests a file block. The applet decrypts the file block, and all subsequent file blocks, with KFB=K(K(FB)) until the applet receives and decrypts all blocks of the file.

The user can forward documents using a forward document form on the system. The form includes a text field in which the user provides information about the file being forwarded, a recipient field (one-click enabled) that can accept multiple subscriber usernames or non-subscriber e-mail addresses, a forward initiation object (such as a button), and an abort object (such as a cancel button).

The system allows users to view a history of documents they have manipulated with the system. The information the system provides preferably includes document name, date of transfer, document size, type of operation, sender name, and recipient name. Viewing the history allows users to detect unauthorized transfers if someone has hijacked their accounts and to keep track of the number of transfers made as compared to the users' limits. Users preferably can neither delete any records from the history nor delete the history itself.

The system notifies a recipient of an incoming secure document by system notification and universal inbox. Non-subscribers preferably receive an e-mail message with a hot link to a particular web page including entrance to the system.

Optionally, the system can notify the sender when a recipient opens a sent file or document. The sender preferably receives an e-mail message stating that the recipient opened the file and is given the option to prevent notification of such occurrences in the future. Once the file has been opened, the sender cannot "unsend" it.

I prefer to provide only a paid access level at which a user is allowed unlimited file transfers. However, other access schemes could be used, such as a scheme including two levels of user privilege: Free and Subscribed. Free users would be allowed particular secure downloads per month, after which additional downloads would count as document transfers. Free users would also have access to a given file for a particular number of days, after which time the system deletes the file. Further, Free users could download up to a particular size limit per download and up to a particular number of transfers per month. Subscribers would

receive more downloads per month, could have access to documents for a longer period, could have a higher size per transfer limit, and could have an unlimited number of transfers per month. In any case, the system deletes documents to which no users have access, which deletion (or "Cleanup") is performed on a monthly basis, checking documents for time restrictions and counters for downloads/transfers, all of which are reset.

My invention can be varied in many ways without exceeding the scope of the inventive concept. For example, ECC can be used to generate the session key and Triple DES can be used to encrypt and decrypt the file. We could also use a variety of symmetrical encryption algorithms for encryption, including Rijndael, Blowfish, and future algorithms developed for the Advanced Encryption Standard.